Professional Web APIs with PHP: eBay®, Google®, PayPal®, Amazon®, FedEx®, plus Web Feeds

Paul Reinheimer

Wiley Publishing, Inc.

Professional Web APIs with PHP: eBay®, Google®, PayPal®, Amazon®, FedEx®, plus Web Feeds

Professional Web APIs with PHP: eBay®, Google®, PayPal®, Amazon®, FedEx®, plus Web Feeds

Paul Reinheimer

Wiley Publishing, Inc.

Professional Web APIs with PHP: eBay®, Google®, PayPal®, Amazon®, FedEx®, plus Web Feeds

Published by
Wiley Publishing, Inc.
10475 Crosspoint Boulevard
Indianapolis, IN 46256
www.wiley.com

Published by Wiley Publishing, Inc., Indianapolis, Indiana

Published simultaneously in Canada

ISBN-13: 978-0-7645-8954-6
ISBN-10: 0-7645-8954-7

Manufactured in the United States of America

10 9 8 7 6 5 4 3 2 1

1B/RS/QX/QW/IN

About the Author

Paul Reinheimer (Ontario, Canada) is a Zend Certified Engineer and principal member of the PHP Security Consortium (PHPSEC). He is currently developing Share The Beat (`www.sharethebeat.com`), an online music portal developed extensively in PHP. Paul has been programming in PHP for the last three years on a multitude of professional and personal projects.

Credits

Acquisitions Editor
Kit Kemper

Development Editor
Ed Connor

Technical Editor
Chris Shiflett
David Mercer

Production Editor
Pamela Hanley

Copy Editor
Kim Cofer

Editorial Manager
Mary Beth Wakefield

Production Manager
Tim Tate

Vice President and Executive Group Publisher
Richard Swadley

Vice President and Executive Publisher
Joseph B. Wikert

Graphic and Production Specialists
Jennifer Click
Joyce Haughey
Alicia B. South
Julie Trippetti

Quality Control Technician
Jessica Kramer

Project Coordinator
Ryan Steffen

Proofreading and Indexing
Techbooks

For my loving parents

Acknowledgments

It wasn't until I started writing that I realized that the name on the front cover of a book is just the tip of the iceberg when it comes to significant contributions. Thanks to Chris, David, Debra, Ed, Kit, and Scott for not losing faith in me when my first chapters came back more wrong than right, and for helping me find "right" each step of the way. And a very special thanks to Laura for supporting me throughout this journey.

Contents

Introduction

With the growing popularity of the Internet, not merely as a research tool or toy, but as a bona fide business communication tool, groups are finding new ways to communicate with each other. Initially (mirroring the vendor-based communication protocols that ran rampant before standardized open protocols were introduced) they communicated via proprietary closed protocols. Fortunately, technologies like XML feeds and broader web services are allowing communication frameworks to be built faster, and allowing more groups to participate.

In this book, feeds are introduced first. These XML documents are used to pass information off from one party to others. These feeds are frequently used by news sites (both professional and amateur) to pass off their stories to interested third parties. Feeds are frequently used to mirror content available on the general website. By providing this same information in a convenient XML format, users are able to easily integrate it into their own site without resorting to cumbersome (and often unreliable) scraping techniques. Both aspects of feeds are discussed—producing the feeds to provide your users with your content in an easy-to-use format, and consuming those feeds to present external content to your users.

Second, APIs are introduced. Whereas feeds provide the same document to all requestors, the response an API provides is very dependent on the requestor and the specifics of the request. Allowing the user to request specific information opens a whole new world of opportunities, where detailed information can be requested on anything the server offers, or frequently, to push information to the server itself. APIs often allow users to connect to the server via a secure channel, which allows confidential transfers such as money transfers or bidding on auctions. A series of existing APIs are presented, complete with working code.

Although these topics are nothing new to the bookseller's shelf, I have often been frustrated with the common approach of exploring a single problem in a variety of languages. As a PHP programmer, I read the PHP sections and skip the rest. This leaves me paying for a whole book, but only reading a quarter of it. While you may have bought this book with a specific API or project in mind, my hope is that by covering a variety of things in a single language, you will not only find a more detailed coverage of that specific topic, but will also find other topics of interest, which you can hopefully use later.

Who This Book Is For

This book was written with the beginner to intermediate PHP programmer in mind, and as such a good understanding of PHP is assumed throughout the book. That being said, complicated concepts or code examples that make use of PHP's more arcane features are explained carefully.

Both feeds and APIs communicate primarily in XML, so a good understanding of XML would be beneficial. XML is introduced and discussed in Chapter 2; you may want to read that section before proceeding elsewhere if you have not previously examined XML.

How This Book Is Structured

This book has two sections, Web Feeds and APIs. Logically enough the first section discusses web feeds, from both the production and consumption angles. As such, the section is divided into three chapters, covering the introduction, consumption, and production of web feeds. The second section examines APIs. Past the introductory chapter there are four chapters, each delving into a single API. Following that there is a chapter that briefly examines three other APIs, followed by a chapter that examines APIs from the other side, producing an API to offer to your users. Each section and chapter stands on its own, with few exceptions. XML is introduced in Chapter 2, as are the basic formats for web feeds. The basic structure for REST and SOAP APIs are introduced in Chapter 5, useful things to know while working with APIs. For both the feeds and API sections, I would recommend reading at least one of the chapters that concentrates on consumption before moving onto the production side of things. This will help ensure you have a good grounding

What You Need to Use This Book

The code in this book was written for PHP5, and it is strongly recommended that you upgrade to PHP5 if you haven't already. That being said, with the exception of the examples using `SimpleXML()` the code will run under PHP4 (and even in those cases, similar functionality is available from third-party libraries). The examples in this book that require a database use either MySQL or SQLite, so having them available on your test machine would be a good idea (PHP5 includes SQLite in a default install).

Unfortunately, PHP doesn't ship with all of the things required for some of the examples in this book. Appendix C outlines all of the options PHP was built with to develop the code shown within these pages.

Finally, a few of the examples in this book make use of some of the tools that come with the OpenSSL package under Linux to create certificates for authentication purposes. If you do not have access to a machine with this package, you will need to find another way to create these certificates.

Conventions

To help you get the most from the text and keep track of what's happening, we've used a number of conventions throughout the book.

Boxes like this one hold important, not-to-be forgotten information that is directly relevant to the surrounding text.

Tips, hints, tricks, and asides to the current discussion are offset and placed in italics like this.

As for styles in the text:

- We *highlight* new terms and important words when we introduce them.
- We show keyboard strokes like this: Ctrl+A.
- We show file names, URLs, and code within the text like so: `persistence.properties`.
- We present code in two different ways:

```
In code examples we highlight new and important code with a gray background.

The gray highlighting is not used for code that's less important in the present
context, or has been shown before.
```

Source Code

As you work through the examples in this book, you may choose either to type in all the code manually or to use the source code files that accompany the book. All of the source code used in this book is available for download at `http://www.wrox.com`. Once at the site, simply locate the book's title (either by using the Search box or by using one of the title lists) and click the Download Code link on the book's detail page to obtain all the source code for the book.

> *Because many books have similar titles, you may find it easiest to search by ISBN; for this book the ISBN is 0-7645-8954-7.*

Once you download the code, just decompress it with your favorite compression tool. Alternatively, you can go to the main Wrox code download page at `http://www.wrox.com/dynamic/books/download.aspx` to see the code available for this book and all other Wrox books.

Errata

We make every effort to ensure that there are no errors in the text or in the code. However, no one is perfect, and mistakes do occur. If you find an error in one of our books, like a spelling mistake or faulty piece of code, we would be very grateful for your feedback. By sending in errata you may save another reader hours of frustration and at the same time you will be helping us provide even higher quality information.

To find the errata page for this book, go to `http://www.wrox.com` and locate the title using the Search box or one of the title lists. Then, on the book details page, click the Book Errata link. On this page you can view all errata that has been submitted for this book and posted by Wrox editors. A complete book list including links to each book's errata is also available at `www.wrox.com/misc-pages/booklist.shtml`.

If you don't spot "your" error on the Book Errata page, go to `www.wrox.com/contact/techsupport.shtml` and complete the form there to send us the error you have found. We'll check the information and, if appropriate, post a message to the book's errata page and fix the problem in subsequent editions of the book.

p2p.wrox.com

For author and peer discussion, join the P2P forums at `p2p.wrox.com`. The forums are a Web-based system for you to post messages relating to Wrox books and related technologies and interact with other readers and technology users. The forums offer a subscription feature to e-mail you topics of interest of your choosing when new posts are made to the forums. Wrox authors, editors, other industry experts, and your fellow readers are present on these forums.

At `http://p2p.wrox.com` you will find a number of different forums that will help you not only as you read this book, but also as you develop your own applications. To join the forums, just follow these steps:

1. Go to `p2p.wrox.com` and click the Register link.
2. Read the terms of use and click Agree.
3. Complete the required information to join as well as any optional information you wish to provide and click Submit.
4. You will receive an e-mail with information describing how to verify your account and complete the joining process.

You can read messages in the forums without joining P2P but in order to post your own messages, you must join.

Once you join, you can post new messages and respond to messages other users post. You can read messages at any time on the Web. If you would like to have new messages from a particular forum e-mailed to you, click the Subscribe to this Forum icon by the forum name in the forum listing.

For more information about how to use the Wrox P2P, be sure to read the P2P FAQs for answers to questions about how the forum software works as well as many common questions specific to P2P and Wrox books. To read the FAQs, click the FAQ link on any P2P page.

Part I
Web Feeds

1

Introducing Web Services

I imagine you picked up this book for a reason.

Perhaps you've built a fantastic website for your organization, but now that it's done, you want more: more ways to view the information, more ways to alter the information, more ways to share the information. Web services can help.

Maybe you run a small business through eBay and want better ways to manage the items you are selling. Web services can help.

You could be a student, a teacher, or an institution wanting to leverage the vast wealth of information compiled by Google. Web services can help.

Web services can help you use the information on the Web in new ways, from simply allowing someone to subscribe to pockets of information so that they can be immediately notified when new information is available, to creating complex business applications that can manage complex real-time supply chains.

Before digging in to how to use web services in the following chapters, this chapter presents you with the following:

- What web services are
- Why web services are used
- How web services work
- Who uses web services
- And finally, why you should use web services

Defining Web Services

Web services are a collection of protocols that are used to exchange data between disparate applications or systems. The essence of web services is the open standards on which they are built, by leveraging public and common protocols like HTTP, along with the XML document model. Web services are easy to implement with existing technologies. Not only are you (as a PHP developer) already familiar with many aspects of web services and the tools required, but you already have the facilities required to deploy them.

Why They Are Used

Simply put, web services allow information to easily pass from those who have it to those who desire it. Both feeds and APIs allow the requestor to obtain information from the service, regardless of the information type. Web services allow customers to access the information they desire, without the extraneous information generally presented on web pages. For example, running a search for a book on Amazon.com will yield several relevant results, generally (under the current layout) in the center column of the page. The right and left columns will contain other information, not directly related to your search (recommended titles, recently viewed items, and so on). Performing the same search via the API will yield the same results, but without that extraneous information.

> *This points to a key factor in web services — they are used to obtain specific information, or complete specific tasks. Unlike people, the automated processes that utilize web services don't make impulse decisions (like choosing a recommended title from Amazon), so there is no point in presenting them with those types of options.*

How They Are Used

Web services generally present information already available via another method (that is, a website). The advantage in the services is the consistent presentation of the information (in stark contrast to methods like screen scraping) in an easy-to-parse format. Here are some examples:

- Federal Express — Allows customers to do rate lookups and schedule shipments after logging into its online system. These same tasks can be accomplished via its API; as such, customers with larger shipping needs can integrate the API with their own systems to quickly automate shipping, and allow customers to determine how much it will cost to ship a particular product.
- Amazon — Allows customers to search for products via its website, and similar tasks can be accomplished via its API (often with more granularity than the traditional web interface provides). Using the API allows small booksellers to transparently integrate with Amazon to offer additional books and apply their own pricing.
- My Personal Blog — Allows people to read my thoughts on various professional matters. The feeds provided by my blog allow those people to read the posts on their own terms, within their own client, without ads or cute pictures of my cat.

It's important to recognize both types of interaction presented here. In some cases the web service is accessed directly by the end user (as is often the case with blogs and feeds). In other cases the service is consumed by an intermediate service, then presented (through various means) to its users.

Who Uses Them

If you accept my broad definition of web services, including both feeds and APIs, the answer is almost everyone. Most news sites offer at least some of their information via an XML feed. Blog sites almost without exception include some form of feed, and APIs are becoming more prevalent offerings from businesses of all sizes.

Also keep in mind that the scope of web services varies widely depending on who offers them. Many are merely informational in nature (such as the National Weather Service API), providing read-only access to information. Others, however, allow you to present information to the server; this could be purely digital in nature (such as adding a bookmark to your del.icio.us account), or it could set into motion a series of physical events (like scheduling a package pickup through FedEx).

Why You Should Use Web Services

There are two main reasons to use web services, both remarkably straightforward: someone else has information you require, or you have information you wish to provide to others. When planning a new web service, remember that servers can either provide or accept information. Remember that web services can provide any level of security, from a completely open service open to all who request it, to a private service with transfers happening over SSL and client-side certificates to validate identities.

Convincing the Boss

Here are a few more tidbits that should help convince the boss (or you) that web services are a route to look into:

- Web services make use of technology you already have deployed — you already have a web server, you already have a database, and so on. Why not provide another way to access the data?
- Web services fit into the security scheme you are already employing. Careful deployment will keep security concerns in line with regular web access.
- Technically savvy users will find a way to access the information they desire. Without web services they will need to resort to methods like screen scraping, which is more difficult to control and can be unreliable (and providing unreliable access to your information is worse than providing none at all).
- Web services can help manage costs; easy integration with companies (like FedEx) that offer services to you can result in lower variable costs for your transactions.
- Web services will allow your business to scale; allowing clients to integrate their systems with yours will help encourage continued relationships, and avoid devoting resources to creating systems for specific clients.

Summary

This chapter answered some basic questions about web services that should allow you to do the following:

- Explain what web services are and how they work
- See the value in utilizing web services in your own projects
- Convince others that it's definitely worth the effort to build and use web services

In the next chapter, you start working with web feeds, a simple but useful form of web services. As you progress through the book, you'll start getting into more complex uses of web services and eventually create your own web service APIs.

2

Introducing Web Feeds

Web feeds (or simply feeds) are a quickly growing, easy-to-deploy technology that allows both providers and users of the feed easy access to the relevant information. Feeds are regularly updated eXtensible Markup Language (XML) documents, generally containing some basic information about the site or group offering the feed, followed by a group of items containing whatever information the feed producers want to disseminate.

Of course, there are a whole host of different uses for web feeds. For example, you could stick a work-related feed onto your intranet site to keep employees up to date on industry news, or you could combine the feeds from various blogs and websites to create a personal start page containing all the sites you read in a condensed format. Just find the feeds you're interested in and away you go!

Obviously, having the ability to view the latest information, in a given subject area and in an easily readable format, can be a huge bonus for a host of businesses and people. Accordingly, this chapter looks at the following:

- What web feeds are
- Important considerations when using feeds
- Key differences between HTML and XML
- Who provides web feeds
- What software uses web feeds

You might be asking yourself whether there are other ways to get information from the Web. Well, in general, feeds are the only reliable (and legal) way to obtain many types of information from other websites in an automated manner. Other methods, such as scraping HTML pages, are not only frowned upon, but may be illegal (if you choose to present this information to others).

What Are Feeds?

You can think of feeds as small modules of information that can be plugged into existing websites, consumed by clients on their desktop, or consumed by aggregators to be presented by users with other feeds. Aggregators also offer searching functionality to users, allowing new users to locate your site and feed (a great reason to provide a feed in the first place).

- Websites such as Yahoo! produce web feeds.
- Software that downloads and uses feeds is said to consume or aggregate feeds.
- Sites such as Google News that retrieve feeds from a number of sources and display selected items are called aggregators.

Figure 2-1 shows feeds in action.

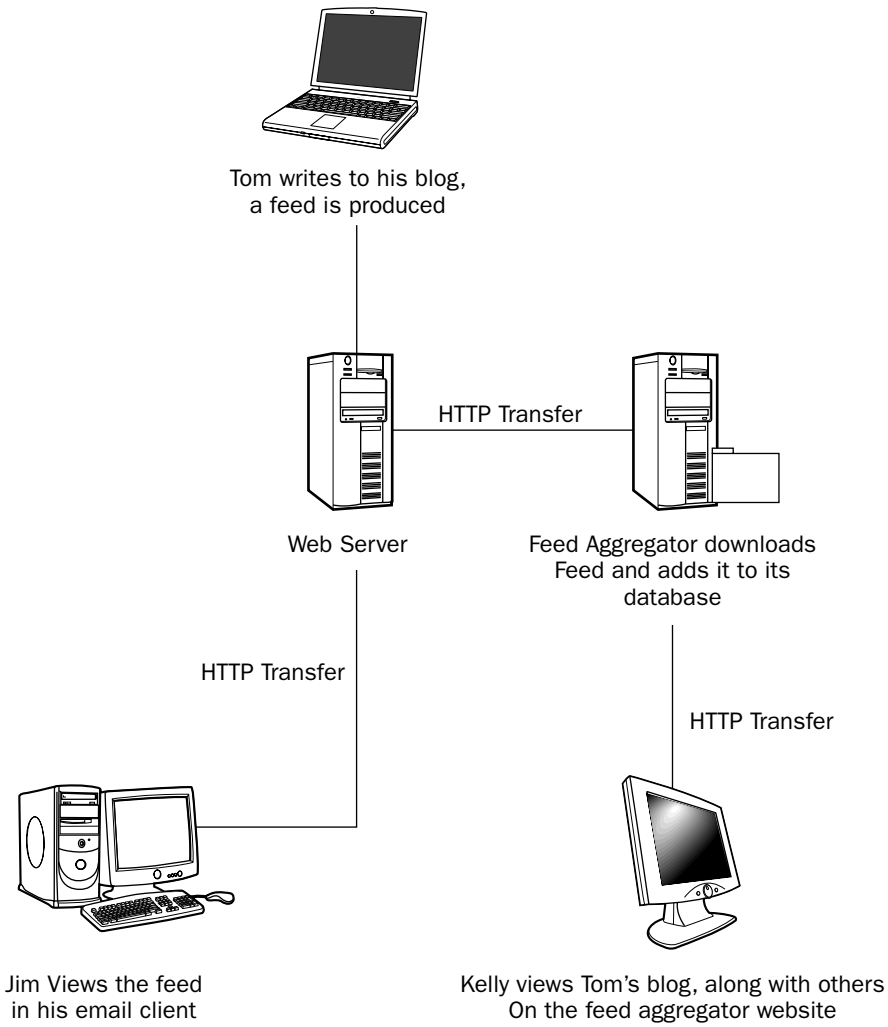

Figure 2-1